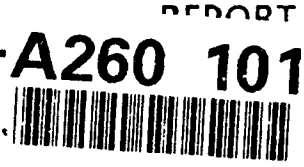


REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

12

2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
February 1992	memorandum

4. TITLE AND SUBTITLE

Data and Model-Driven Selection using Color Regions

5. FUNDING NUMBERS

DACA76-85-C-0010
N00014-85-K-0124
IRI-8900267

6. AUTHOR(S)

Tanveer Fathima Syeda-Mahmood

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

AIM 1270

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Information Systems
Arlington, Virginia 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution of this document is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A key problem in model-based object recognition is selection, namely, the problem of determining which regions in the image are likely to come from a single object. In this paper we present an approach that uses color as a cue to perform selection either based solely on image-data (data-driven), or based on the knowledge of the color description of the model (model-driven). Specifically, the paper argues for the specification of color in terms of color categories as being appropriate for the task of selection. These color categories are used to develop a fast region segmentation algorithm that extracts perceptual color regions in images. The color regions extracted form the basis for performing data and model-driven selection. Data-driven selection is achieved by

(continued on back)

14. SUBJECT TERMS (key words)

selection, recognition, visual attention, color, saliency, region segmentation

15. NUMBER OF PAGES

29

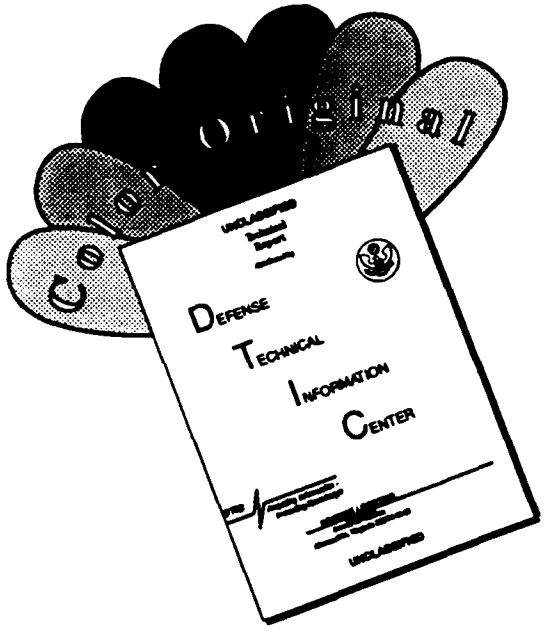
16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

Block 13 continued:

selecting salient color regions as judged by a color-saliency measure that emphasizes attributes that are also important in human color perception. The approach to model-driven selection, on the other hand, exploits the color region information in the model to locate instances of the model in a given image. The approach presented tolerates some of the problems of occlusion, pose and illumination changes that make a model instance in an image appear different from its original description. Finally, the utility of color-based data and model-driven selection is discussed in the context of reducing the search involved in recognition.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

(12)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1270 February 1992

Data and Model-Driven Selection using Color Regions

Tanveer Fathima Syeda-Mahmood

Abstract

A key problem in model-based object recognition is selection, namely, the problem of determining which regions in the image are likely to come from a single object. In this paper we present an approach that uses color as a cue to perform selection either based solely on image-data (data-driven), or based on the knowledge of the color description of the model (model-driven). Specifically, the paper argues for the specification of color in terms of color categories as being appropriate for the task of selection. These color categories are used to develop a fast region segmentation algorithm that extracts perceptual color regions in images. The color regions extracted form the basis for performing data and model-driven selection. Data-driven selection is achieved by selecting salient color regions as judged by a color-saliency measure that emphasizes attributes that are also important in human color perception. The approach to model-driven selection, on the other hand, exploits the color region information in the model to locate instances of the model in a given image. The approach presented tolerates some of the problems of occlusion, pose and illumination changes that make a model instance in an image appear different from its original description. Finally, the utility of color-based data and model-driven selection is discussed in the context of reducing the search involved in recognition.

32

This work is supported by an IBM Graduate Fellowship. It describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010, under Office of Naval Research contract N00014-85-K-0124 and under NSF Grant IR1-8900267.

1. SELECTION IN RECOGNITION

A key problem in object recognition is selection, namely, the problem of identifying regions in an image within which to start the recognition process, ideally by isolating regions in an image that are likely to come from a single object. Model-based object recognition methods that try to recognize which members of their library of models are present in the scene, usually use geometric features such as points or edges and try to identify pairings between data and model features that are consistent with a rigid transformation of the object model into image coordinates. The large number of such pairings that need to be examined in cluttered scenes leads to a combinatorially explosive search problem. It has been shown that this search can be considerably reduced if recognition systems are equipped with a selection stage where subsets of data features can be isolated that are likely to come from a single object, thus allowing the search to be focused on those matches that are more likely to lead to a correct solution [12]. This isolation can be either based solely on image data (data-driven) or can incorporate the knowledge of the model (task-driven or model-driven). In addition, it is desirable to order these subsets of data features such that the more promising ones, i.e., those that are more likely to point to a single object, are explored first. This can not only increase the likelihood of a good match being obtained earlier, but is also useful when the task is to recognize as many objects as possible in a scene. Thus the goals of selection in recognition are two-fold: To isolate areas in the image that are likely to come from a single object, and to order these regions such that the more promising ones are explored first. These goals of selection are different from those of segmentation, where the problem is to partition the image into regions that contain a single object. In selection, on the other hand, it is not essential to isolate regions that totally contain a single object, nor is it necessary to partition the entire image into different object containing regions.

Even though selection can be of help in recognition, it has largely remained unsolved. What makes selection so difficult? In the ideal case, if the appearance of the desired object in the scene were known, and objects in the scene were nicely separated and distinguishable from the background, and the illumination conditions were known, then even simple methods that rely on intensity measurements would work well to extract groups of features. But in reality, the appearance of the object is not known. In addition, illumination conditions and surface geometries of objects present in a scene can cause problems of occlusion, shadowing, specularities, and inter-reflections in the image and make it difficult to interpret groups of data features such as edges and lines. Previous approaches to selection have focused on the problem of data-driven selection by grouping data features such as edges, lines, points, or based on constraints such as parallelism, or collinearity, [19], distance and orientation [18], and regions enclosed by a group of edges [6]. The extent to which such grouping methods reduce the search in recognition depends

on the reliability of the groups produced (i.e. how many of them really come from a single object). Maintaining the reliability of groups was found to be difficult using constraints such as the ones listed above. So the general problem of selection remains largely unsolved as it is still not obvious how to reliably characterize subsets of data features that will give clues that point to a single object. Thus it appears that there is a need for a computational model of selection to explain both data and task-driven selection.

We have been involved in building one such model that proposes that selection be accomplished via an attention mechanism. Specifically, it is an attempt to build a computational model of the visual attention phenomenon in humans, and to propose it as a selection mechanism for recognition. This involves the isolation of two modes of human attentional behavior, namely attracted-attention and pay-attention modes, to serve as paradigms for data-driven and model-driven selection respectively. The *attracted-attention* mode of behavior is spontaneous and is commonly exhibited by an unbiased observer (i.e., with no *a priori* intentions) when some objects or some aspects of the scene attract his/her attention. The *pay-attention* mode is a more deliberate behavior exhibited by an observer looking at a scene with *a priori* goals (such as the task of recognizing an object, say) and hence paying attention to only those objects/aspects of a scene that are relevant to the goal. According to this model, therefore, data-driven selection can be achieved by identifying regions in an image that attract attention (i.e., that are distinctive) with respect to some feature such as color or texture, while model-driven selection can be achieved by paying attention to the model features (i.e., using the model features to decide saliency of features in the image). While it is understandable that paying attention to model features can help isolate areas in the image that could contain subsets of data features that are likely to contain a single object (or the specific model object in this case), it is not immediately apparent how locating salient regions can help in serving the goals of selection. Such a choice is, however, motivated by the following considerations. First, it is often observed that an object stands out in a scene because of some distinctive features that are usually localized to some portion of the object. Therefore isolating distinctive regions is more likely to point to a single object. Secondly, a distinctive region, if suitably found, can help in limiting the number of candidate models from the library that can potentially match the given data. This is especially true if only a few models in the library satisfy the features that made the data region distinctive. Lastly, it has often been observed that the first objects recognized in a scene are those that attract an observer's attention [15]. Thus ordering the regions by distinctiveness to decide which objects to recognize first seems to be in keeping with this observation. Finally, a number of other approaches have also suggested that selection, at least data-driven, can be performed based on some measure of saliency, such as the structural saliency of curves [29], or saliency defined by local differences in

contrast, color, or size [8, 24, 28].

The above discussion indicates a framework in which data and model-driven selection can be achieved. But how can salient regions be found in the image independent of the model, and how can the object model affect the choice of regions? The purpose of this paper is to present a method of selection by restricting attention to one particular feature, namely, color. It shows how color regions can be extracted from the image and how they can be used to perform data-driven and model-driven selection. To give a flavor for the ensuing discussions, Figures 1-3 show some examples of the results of data and model-driven selection performed by our system. Figure 1a shows an image of a realistic indoor scene with shadows, inter-reflections, and consisting of many types of objects. The different color regions found in this image are re-colored and shown in Figure 1b. The four most salient color regions found are shown in Figures 1c-1f. These regions span objects in the scene that are salient in color. Figure 2-3 show model-driven selection using color, using the model object shown in Figure 2a and the scene depicted in Figure 2c. The cluster of regions found to best satisfy the model color region description using our algorithm for model-driven selection is shown in Figure 3d.

The rest of the paper discusses how this kind of selection can be achieved using color. It is organized as follows. In Section 2, we motivate the choice of color as a feature to study selection, and outline the requirements imposed by selection on any method for the extraction of color information. Based on these guidelines, an approach to extracting color regions is presented in Section 3. In Section 4, a measure for expressing the saliency of color regions is presented and its effectiveness for data-driven selection is examined. Section 5 presents a way to perform model-driven selection based on the color regions. Finally, Section 6 summarizes our approach to color-based selection.

2. COLOR IN SELECTION

2.1 Role of Color in Selection

Color is known to be a strong cue in attracting an observer's attention. Humans often also use color information to search for specific objects in a scene. It therefore seems natural to use color as a cue for performing selection in computer vision. But the strong motivation for using color in selection comes from the fact that it provides region information and that, when specified appropriately, it can be relatively insensitive to variations in normal illumination conditions and appearances of objects [31]. A color region in the image almost always comes entirely from a single object, giving, therefore, more reliable groups than existing grouping methods and this can be useful for data-driven selection. Because objects tend to show color constancy under most illumination conditions, color can be a stable cue for most poses (appearances) of objects in scenes, thus making it also suitable for model-driven selection.

2.2 Surface Color, Image Color, Perceptual Color

Although color is useful for selection, the problem of specifying the *perceived color* of objects, that is, the color perceived by humans looking at an image of the scene, has proven to be difficult in computer vision. Several artifacts such as specularities (from shiny surfaces in the scene), inter-reflections, shading on surfaces, and shadowing all make it difficult to recover the actual color of objects in the scene from the image. Existing approaches have mainly focused on the problem of color constancy, where the goal was to extract surface color, i.e., surface reflectance properties of objects, in order to obtain a stable perception of the color of an object under varying illumination conditions. As this problem is under-constrained, most methods make some assumptions about either the surface being imaged [23], or about the illumination conditions [25, 14, 11, 32], or both [10]. Other approaches also exist that try to recover *image color*, i.e., the color of the objects as they appear under the present illumination conditions, accounting separately for artifacts such as specularities on shiny surfaces [22]. These methods, however, cannot ensure that the color extracted matches the perceived color of regions.

For the purposes of selection, what kind of color information should be extracted from regions? Is recovering image color sufficient or should one attempt to recover surface color? We propose that for both data and model-driven selection, it is sufficient if a region could be specified by its perceived color, and the effects of artifacts such as specularities could be separately accounted as was done by image color recovery methods. Using the perceptual color, two adjacent color regions would be distinguished if their perceived colors were different, and this is sufficient for data-driven selection. Because objects tend to obey color constancy under most changes in illumination, their perceived color remains more or less the same thus making it sufficient also for model-driven selection. But can perceptual color be quantified at all? In general, several effects such as simultaneous color contrast and color filling, have been known to influence human perception of color [34]. Fortunately, (as we will explain later,) these factors are not very critical for selection.

2.3 Perceptual Color Specification by Categories

In this section we present a method for specifying the perceptual color of image regions from the colors of their constituent pixels. The color of pixels in images is described by a triplet <R,G,B> (called *specific color* henceforth), representing the components of image intensity at that point along three wavelengths (usually red, green and blue as dominant wavelengths to correspond to the filters used in the color cameras). When all possible triples are mapped into a 3-dimensional color space with axes standing for the pure red, green and blue respectively, we obtain a color space that represents the entire spectrum of computer recordable colors. Such a color space must, therefore, be partitionable into subspaces where the color remains perceptually the same, and is distinctly different from that of

neighboring subspaces. Such subspaces can be called *perceptual color categories.* Now each pixel in the image maps to a point in this color space, and hence will fall into one of these categories. *The perceptual color of this pixel can, therefore, be specified by this color category.* To obtain the perceived colors of regions from the perceptual color of their constituent pixels, we observe the following. Although the individual pixels of an image color region may show considerable variation in their specific colors, the overall color of the region is fairly well-determined by the color of the majority of pixels (called *dominant color* henceforth). *Therefore, the perceived color of a region can be specified by the color category corresponding to the dominant color in the region.*

The category-based specification of perceptual color (of pixels or regions) is a good compromise between choosing the specific color (which is extremely unstable with respect to changes in illumination conditions, etc.) and surface color (whose recovery is hard). Since the categories indicate the perceptual color, they have the same beneficial effect as recovering perceptual color, on both data and model-driven selection, such as giving a reliable segmentation of image into color regions, and being stable under changes in illumination conditions. In addition, since the perceptual categories depend on the color space and are independent of the image, they can be found in advance and stored in, say, a look-up table. Finally, a category-based description is in keeping with the idea of perceptual categorization that has been explored extensively through psychophysical studies [4, 5, 27]. These studies concluded that although humans can discriminate between several thousand nuances of colors, psychophysically, we seem to partition the color space into relatively few distinct qualitative color sensations or categories [30].

2.4 Categorization of Color Space

The above discussion argued for the viability of an approach that recovers a color to within a category. Before this can be turned into a computational method of color recovery one needs to address the issue of how such categories may be found. Previous work on color categorization involved experiments of naming the color using a limited vocabulary, or identifying colors using the Munsell color charts [34]. But for computational color recovery, we need a way to convert the camera recordable red, green and blue components of colors into computer recordable perceptual color categories. This was done by performing some rather informal but extensive psychophysical experiments that systematically examined a color space and recorded the places where qualitative color changes occur, thus determining the number of distinct color categories that can be perceived. For this, the hue-saturation-value color space was used as it specifies a given color in terms of its hue, purity and brilliance - attributes that have been found to give a perceptual description of color [20]. The details of these experiments are described in Appendix A and will not be elaborated here, except to mention the following. The entire spectrum of computer recordable colors (2^{24} colors) was

quantized into 7200 bins corresponding to a 5 degree resolution in hue, and 10 levels of quantization of saturation and intensity values (see Figure 7). The color in each such bin was then observed by displaying a mondrian (a uniform color patch) of that color on a monitor screen and observing it under dark room conditions with appropriate monitor calibration. From our studies, we found about 220 different color categories were sufficient to describe the color space. The color category information was then summarized in a *color-look-up table*. Although it is true that a finer level of quantization would have yielded more categories, a smaller set is actually more useful since it gives a reasonably coarse description of the color of a region thus allowing it to remain the same for some variations in imaging conditions. In fact, by the above method we can also determine which categories can be grouped to give an even rougher description of a particular hue. This was done and stored in a *category-look-up table* to be indexed using the color categories given by the color-look-up table.

3. COLOR REGION SEGMENTATION

The previous section described how to specify the color of regions, after they have been isolated. But the more crucial problem is to identify these regions. In this section, we show that the perceptual categorization principle can be used to determine which pixels can be grouped to form regions in an image. If each surface in the scene were a mondrian, then all its pixels would belong to a single color category, so that by grouping spatially close pixels belonging to a category, the desired segmentation of the image can be obtained. But real surfaces being hardly mondrians, it is rare that pixels of a region from such surfaces all belong to the same color category. They could show considerable variation in color with bright and dark pixels intermixed, and with possibly spurious pixels also being present. We now analyse some of the color variations across an image that can result from imaging a colored surface in the scene.

3.1 Variation of Color Across an Image of a 3D-Surface

In this section we use some assumptions to show that the color variations across an image of a surface is mostly in intensity. When a surface is imaged, the light falling on the image plane (image irradiance) is related to the physical properties of the scene being imaged via the image irradiance equation:

$$I(\lambda, r) = \rho(\lambda, \mathbf{r}) F(\mathbf{k}, \mathbf{n}, \mathbf{s}) E(\lambda, \mathbf{r}). \tag{1}$$

where λ is the wavelength, $\mathbf{r}$ is the spatial coordinate and r is its projection in the image, $E(\lambda, \mathbf{r})$ is the intensity of the ambient illumination, $\rho(\lambda, \mathbf{r})$ is the component of surface reflectance that depends only on the material properties of the surface (and hence specifies its surface color), while $F(\mathbf{k}, \mathbf{n}, \mathbf{s})$ is the component of surface

reflectivity that depends on surface geometry, with **k**,**s**,**n** being the viewer direction, the source direction and the surface normal respectively. Although the image irradiance equation assumes that all surfaces in a scene reflect light governed by a single reflectivity function, we can easily reinterpret this equation to represent image irradiance of a single surface. Under the assumption of a single light source, the surface illumination $E(\lambda, \mathbf{r})$ can be separated as a product of two terms $E_1(\lambda)$ and $E_2(\mathbf{r})$, and since $F(\mathbf{k}, \mathbf{n}\ \mathbf{s})$ is a function of position $\mathbf{r}$ it can be expressed as $\mathcal{F}(\mathbf{r})$. Then the image irradiance equation can be re-written as

$$I(\lambda, r) = \rho(\lambda, \mathbf{r})\mathcal{F}(\mathbf{r})E_1(\lambda)E_2(\mathbf{r}). \tag{2}$$

The surface reflectance and hence the resulting appearance of a surface is determined by the composition as well as the concentration of the pigments of the material constituting the surface. For most surfaces, the composition of the pigments can be considered independent of their concentration so that the spectral reflectance $\rho(\lambda, \mathbf{r})$ can be written as a product of two terms $\rho_1(\lambda)$ and $\rho_2(\mathbf{r})$. Note that this assumption is less restricting than the assumption of homogeneity that has been used before [14]. With this simplification, (and grouping the product of terms dependent on λ and $\mathbf{r}$ separately) the image irradiance equation becomes

$$I(\lambda, r) = H(\mathbf{r})L(\lambda). \tag{3}$$

Now, if we consider the filtered version of this signal, i.e., the image irradiance in three channels, say the red, green and blue channels with their associated transfer functions $h_R(\lambda), h_G(\lambda), h_B(\lambda)$, the specific color at each pixel location r is specified by the triple <R(r),G(r),B(r)> where

$$R(r) = \int_0^\infty I(\lambda, r)h_R(\lambda)d\lambda = H(\mathbf{r})\int_0^\infty L(\lambda)h_R(\lambda)d\lambda = H(\mathbf{r})R_1 \tag{4}$$

$$G(r) = \int_0^\infty I(\lambda, r)h_G(\lambda)d\lambda = H(\mathbf{r})\int_0^\infty L(\lambda)h_G(\lambda)d\lambda = H(\mathbf{r})G_1 \tag{5}$$

$$B(r) = \int_0^\infty I(\lambda, r)h_B(\lambda)d\lambda = H(\mathbf{r})\int_0^\infty L(\lambda)h_B(\lambda)d\lambda = H(\mathbf{r})B_1\ . \tag{6}$$

This shows that under the given assumptions (which include non-homogeneous surfaces,) the color of a surface can vary only in intensity. In practice, even when the separability assumption on reflectance is not satisfied, or there is more than one light source in the scene, the general observation is that the intensity and purity of colors are affected, but the hue still remains fairly constant. In terms of categories, this means that different pixels in a surface belong to *compatible categories*, i.e. have the same overall hue but vary in intensity and saturation. Conversely, if we group pixels belonging to a single category, then each physical surface is spanned by multiple overlapping regions belonging to such compatible color categories. These were the categories that were grouped in the category-look-up-table mentioned in Section 2.4. The next section describes how these concepts can be put together to give a color image segmentation algorithm.

3.2 Color Region Segmentation Algorithm

The algorithm for color image segmentation performs the following steps. (1) First, it maps all pixels to their categories in color space. (2) It then groups pixels belonging to the same category, (3) and finally merges overlapping regions in the image that are of compatible color categories.

1. Mapping pixels to categories: This is done by a simple indexing of the color-look-up-table by the color of the pixel specified in terms of its hue, saturation, and brightness components. These components can be derived from the specific color as described in [9]. This step takes time = O(N) where N is the size of the image.

2. Grouping pixels of same category: The image is divided into small non-overlapping bins of fixed size (, say, 8x8) and the color categories found in the bins are recorded. The size of the bin can be chosen based on expectations about the average size of color regions found in natural scenes. Each bin thus has a list of color categories summarizing the pixel color information in the bin. Neighboring bins that contain a common color category can be grouped to give a connected component representing an image region of that color category. Since a bin has several color categories, it belongs to several connected components that overlap. The actual grouping algorithm we used is a sequential non-recursive labeling algorithm that simultaneously assembles all the overlapping connected components using the category description in the bins. This algorithm is an extended version of the labeling algorithm for binary images described earlier [13], and uses the union-find data structure to efficiently merge category labels into connected components taking time = $O(k^2 M)$ where M = number of windows, and k = maximum number of categories present in the window (= O(1) for small window-sizes, eg., 8 x 8). The resulting labels are propagated back to the pixels to give the precise boundaries of color regions of single color categories. The color of the region is then specified by the color category and specific color that is the dominant color in the region as described in Section 2.3.

3. Merging overlapping regions: The general problem of determining which regions overlap in the image can be a computationally intensive operation as it involves determining which polygonal regions intersect and finding their regions of intersection. But by using the bin-wise representation of connected components, we can detect and combine overlapping regions with greater ease. From the discussion in Section 3.1, a shaded region maps to categories in color space that are compatible, i.e., have the same overall hue. The categories that are compatible are available from the category look-up-table described in Section 2.4. To find all such regions that have compatible categories and overlap in image space, the algorithm examines each window of the image to see if it contains the interior portions of regions of compatible color categories. Such overlap regions are grouped as in step 2. This step again takes $O(k^2M)$ time. Finally, the window-level color labels are propagated back to the corresponding pixels to give an accurate localization of the

color region boundaries.

The algorithm for color image segmentation thus makes only a constant number of passes through the image, each being linear in the size of the image.

3.3 Handling Specularities

The above algorithm segments the image into regions according to their perceived color. As we described before, this is sufficient for data-driven selection. But for model-driven selection such a description needs to be augmented with the knowledge of artifacts that occur in the image such as specularities, shadows, or inter-reflections. Such artifacts can cause a model region to appear fragmented. For example, a sharp streak of specularity on the surface can cleave its image into two regions. If these artifacts could be identified and corrected, this can improve the effectiveness of a color-based model-driven selection system. We now discuss how one of these artifacts, namely, specularities, can be handled once the color regions have been isolated. Specularities are present in regions produced by objects in the scene having shiny surfaces, such as metallic objects and dielectrics. These specularities have a central bright portion that appears white in most illumination conditions (bright sunlight, day light, tube light) and tapers off near the specularity boundary merging into the rest of the body color. Such specular regions and their adjacent colored regions when projected into a color space form characteristic clusters such as the skewed T described in [21]. These clusters can, therefore, be analysed to detect and remove highlights using the method described in that paper.

3.4 Results

Figures 4-6 demonstrate the color region segmentation algorithm. Figure 4a shows a 256 x 256 pixel size image of a color pattern on a plastic bag. The folding on the bag and its plastic material together give a glossy appearance in the image as can be seen in the big S and Y. The result of step-2 of the algorithm is shown in Figure 4b, and there it can be seen that the glossy portions on the big blue Y and the red S cause overlapping color regions. These are merged in step 3 and the result is shown in Figure 4c. As can be seen in the figure, the algorithm achieves a fairly good segmentation of the scene for such surfaces. Figure 5 shows another image consisting of colored pieces of cloth with the textured region having several small colored regions within it. The results of the algorithm (Figure 5c) show that even such colored regions can be reliably isolated. Another example (Figure 1) of color region extraction was mentioned earlier in Section 1. Notice in the segmented image of Figure 1b that adjacent objects of the same perceptual color are merged (grey books). This is to be expected because the grouping of regions is based on color information alone.

4. COLOR-BASED DATA-DRIVEN SELECTION

The segmentation algorithm described above gives a large number of color regions. Some of these may span more than one object, while some come from the scene clutter rather than objects of interest in the scene. It would be useful for the purposes of recognition to order and consider only some of these regions so that by isolating data subsets from such regions, the search can be focused on key groups of features thus excluding much of the scene clutter. Based on the rationale given in Section 1, we propose that the color regions be ordered by their saliency, i.e., by how distinctive they appear. The method of color-based selection, therefore, is to extract color regions from the image, order them based on a measure of color-saliency and then select a few most salient regions to be given to any recognition system. In this section we first describe a measure of expressing color saliency, and then examine the utility of salient-region selection in recognition.

4.1 Finding Salient Color Regions in Images

In trying to express distinctiveness, one encounters the question: Is distinctiveness expressible at all? In general, any judgement of distinctiveness has both a sensory and a subjective component. Thus for example, while most of us can perceive brighter colors more easily than duller colors, the judgement of which of two hues of the same brightness and saturation are more salient can be subjective. The aim here is to focus on the sensory component of distinctiveness and hence extract properties of regions that are general enough to be perceived by most observers. Accordingly, we propose that the saliency of a color region be composed of two components, namely, *self-saliency* and *relative saliency*. Self-saliency determines how conspicuous a region is on its own and measures some intrinsic properties of the region, while relative saliency measures how distinctive the region appears when there are regions of competing distinctiveness in the neighborhood.

In order to develop such a measure for color-region saliency one has to ask the following questions: What features in regions determine their saliency? How can they be measured to reflect our sensory judgments? Finally, how can they be combined to give the saliency measure? We now address these questions and derive a measure of color-saliency.

4.1.1 Features used for measuring self and relative saliency

Since the saliency of a color region depends on the region features used, they must be carefully selected. Such features should be: (i) perceptually important, (ii) easily measurable, and (iii) fairly general, to avoid subjective bias.

1. Color: The color of a region is an intrinsic property and affects a region's self-saliency. It is specified by (s(R),v(R)), where s(R) = saturation or purity of the color of region R, and v(R) = brightness, and $0 \leq s(R),v(R) \leq 1.0$. The hue of colors is not considered, to avoid subjective bias.

2. Region size: The size of a region is again an intrinsic property and affects its self-saliency. It is chosen as a feature based on the observation that regions that are either very small in extent, or that are large enough to cover the entire field of view, do not often attract our attention. Also, very large regions can potentially span more than one object, making them unsuitable for selection. The size feature is expressed by the normalized size r(R) = Size(R)/Image-size.

3. Color contrast: The color contrast a region shows with its neighbors affects its relative-saliency. The rationale behind choosing color contrast is that even if a region has an interesting intrinsic color, it may not be distinctive if all its neighbors also have equally interesting colors, unless it shows the greatest contrast. It is difficult to express color contrast in a numerical measure that can account for the variations in an observer's judgement with the conditions of observation, size, shape, and absolute color of the stimuli [34]. In the color contrast measure we chose, we augmented an empirical color difference formula to predict the observed color differences, with the knowledge of the hues of the colors derived from their categorical representation. Specifically, the following difference formula d(C_R,C_T) was used to measure color difference between two color region R and T with specific colors as $C_R = (r_0, g_0, b_0)^T$ and $C_T = (r, g, b)^T$ as:

$$d(C_R, C_T) = \sqrt{(\frac{r_0}{r_0 + g_0 + b_0} - \frac{r}{r + g + b})^2 + (\frac{g_0}{r_0 + g_0 + b_0} - \frac{g}{r + g + b})^2} \quad (7)$$

As this measure does not explicitly take into account the hues of the colors, the color category-based representation is used to ascertain whether the hues of the two regions are different, and then the extent of difference is judged using $d(C_R, C_T)$ in such a way that the contrast between regions of different hue is emphasized. This allows the measure to handle simultaneous color contrast to some extent. The measure is given by c(R,T) below:

$$c(R,T) = \begin{cases} k_1 d(C_R, C_T) & \text{if R and T are of same hue} \\ k_2 + k_1 d(C_R, C_T) & \text{otherwise} \end{cases} \quad (8)$$

where $k_1 = \frac{0.5}{\sqrt{2}}$ and $k_2 = 0.5$, so that $0 \leq c(R,T) \leq 1.0$.

4. Size contrast: The size contrast is a feature for determining relative saliency and is chosen because it determines if a region is mostly in the background or in the foreground. The size contrast of a region R with respect to an adjacent region T is simply the relative size (area) and is given by

$$t(R,T) = min\left(\frac{\text{size(R)}}{\text{size(T)}}, \frac{\text{size(T)}}{\text{size(R)}}\right) \quad (9)$$

Since a region R has several neighboring regions in general, the color contrast $c(R)$ and size contrast $t(R)$ of a region R are measured relative to a *best* neighbor

T_{best} for each region, so that $c(R) = c(R, T_{best})$, and $t(R) = t(R, T_{best})$. T_{best} is the neighboring region that is ranked the highest when all neighbors are sorted first by size, then by extent of surround, and finally by contrast (size or color contrast as the case may be).

4.1.2 Combining features for self-saliency: To determine self-saliency from the chosen features, they are weighted appropriately to reflect their importance. The self-saliency measure chosen emphasizes purer and brighter colors over darker and duller colors by choosing the weighting functions for saturation and brightness as $f_1(\mathrm{s(R)}) = 0.5\mathrm{s(R)}$, and $f_2(\mathrm{v(R)}) = 0.5\mathrm{v(R)}$ respectively. The size of a region is given a non-linear weight to deemphasize both very small and very large regions as they do not often attract our attention. The corresponding weighting function has sharp as well as smoothly rising and falling phases determined by the breakpoints t_1, t_2, t_3, t_4 as shown in Figure 8a and the equation below.[1] Here n stands for the region size r(R).

$$f_3(n) = \begin{cases} -\frac{ln(1-n)}{c_1} & 0 \le n \le t_1 \\ 1 - e^{-c_2 n} & t_1 < n \le t_2 \\ s_2 - c_3 ln(1 - n + t_2) & t_2 < n \le t_3 \\ s_3 e^{-c_4(n-t_3)} & t_3 < n \le t_4 \\ 0 & t_4 < n \le 1.0 \end{cases} \tag{10}$$

where $\mathrm{t_1} = 0.1$, $\mathrm{t_2} = 0.4$, $\mathrm{t_3} = 0.5$, $\mathrm{t_4} = 0.75$, $\mathrm{s_1} = 0.8$, $\mathrm{s_2} = 1.0$, $\mathrm{s_3} = 0.7$, $\mathrm{s_4} = 10^{-3}$ and $c_1 = -\frac{ln(1-t_1)}{s_1}, c_2 = -\frac{ln(1-s_1)}{t_1}, c_3 = -\frac{(s_2-s_3)}{ln(1+t_2-t_3)}, c_4 = -\frac{ln\frac{s_4}{s_3}}{(t_4-t_3)}$ and n = size of region R = r(R).

4.1.3 Combining features for relative saliency:

Once again, the chosen features are weighted appropriately to determine relative saliency. The color contrast is weighted linearly by a function $f_4(\mathrm{c(R)}) = \mathrm{c(R)}$, to emphasize regions showing greater contrast. The relative size is exponentially weighted by a function $f_5(t(R)) = 1 - e^{-12t(R)}$ to favor situations in which a region and its best neighbor have approximately the same size.[2]

4.1.4 Finding self and relative saliency

Once the various features determining self and relative saliency are appropriately weighted, they reinforce each other so that the self and relative saliencies can be given by simple additive combinations of their individual features. The self-saliency of a region R denoted by SS(R) is given as $f_1(\mathrm{s(R)}) + f_2(\mathrm{v(R)}) +$

[1] Such a function along with the thresholds and rates of change was empirically derived from informal psychophysical experiments (whose details will not be elaborated here) performed using color regions of various sizes.

[2] Once again this function was obtained by performing informal psychophysical experiments.

f_3(r(R)). Similarly, the relative saliency of the region R, RS(R) is given by f_4(c(R)) + f_5(t(R)). Finally, the overall saliency of a region R is expressed by a linear combination of self and relative saliency as SS(R) + RS(R), using the following rationale. Any combination method should be flexible enough to allow a region to be declared salient if it shows good contrast (i.e., high relative saliency) even though it may not be interesting on its own. Conversely, a region that is interesting on its own but fails to become interesting in the presence of neighboring regions should not be chosen. On the basis of these observations alone, nonlinear combining methods such as (SS(R) * RS(R)) or max(SS(R), RS(R)) are not suitable. If a region is both interesting on its own as well as in the presence of other regions in the scene, then it must be given more importance. All three criteria are satisfied when the two saliency components are linearly combined. The color saliency of a region R is therefore given by

$$\text{Color-saliency}(R) = f_1(s(R)) + f_2(v(R)) + f_3(r(R)) + f_4(c(R)) + f_5(t(R)). \quad (11)$$

The saliency measure described above does not completely take into account all the perceptual effects of simultaneous color contrast, color-filling, etc. Because such effects do not greatly undermine a region that is already very outstanding (very salient), and because saliency is being used to rank the regions, we have ignored these effects.

The color regions in the image can now be ordered using the saliency measure and a few most significant regions can be retained for selection (called salient regions, henceforth). The number of salient regions to be retained can be determined when the selection mechanism is integrated with a recognition system to perform a specific task, and is therefore left unspecified here.

4.1.5 Results

We now illustrate the ranking of regions produced by the color saliency measure derived above. Figures 1c-1f show the four most distinctive regions found by applying the color-saliency measure to all the color regions extracted from the scene shown in Figure 1a. Figures 4d-4f, 5d-5f, 6c-6f, show the few most salient regions found in their respective scenes. In the experiments done so far, the color-saliency measure was found to select fairly large bright-colored regions that showed good contrast with their neighbors, and appeared perceptually significant.

4.2 Use of Salient Color-based Selection in Recognition

Data-driven selection based on salient color regions is primarily useful when the object of interest has at least one of its regions appearing salient in the given scene. In such cases, the search for data features that match model features can be restricted to the salient regions, thus avoiding needless search in other areas of the image. By selecting salient color regions, we obtain a small number of groups (a region is itself a group), containing several features. It was shown in [7] that

such large-sized groups are useful for indexing, i.e., to determine which regions from models in a library could correspond to a given group. But when the task is to recognize a single object, it is desirable to have small-sized groups. For this, existing grouping techniques can be applied to the data features found within the color regions to obtain reliable small-sized groups.

We now estimate the search reduction that can be achieved with such a selection mechanism. Let (M,N) = total number of features (such as edges, lines, etc.) in the model and image respectively. Let (M_R, N_R) = total number of color regions in the model and image respectively. Let N_S = number of salient regions that are retained in an image. Let g = average size of a group of data features, within a model or image. Let (G_M, G_N) = number of groups formed (using any existing grouping scheme) in the model and image respectively. Finally, let G_{N_i} be the number of groups in the salient image region i. Using the alignment method of recognition [16], at least three corresponding data features are needed to solve for the pose (appearance) of the model of a rigid object in the image. If no selection of the data features is done, then the brute-force search required to try all possible triples is $O(M^3N^3)$. If selection is done by only grouping methods (i.e., without color region selection), then the number of matches that need to be tried is $O(G_M G_N g^3 g^3)$ since only triples within groups need to be tried. But as we mentioned before, grouping methods often make mistakes, so that not all groups contain features belonging to a single object. In at least one such study [6] out of the 150 or so groups isolated, about 83 groups actually came from single objects. Most of the remaining 67 groups would not yield any consistent match and would represent fruitless search. Consider the case when grouping of data features is done within all the color regions. With this, the grouping is more reliable, and also, the number of groups is smaller (as groups straddling regions are not considered), so that the overall effect is to reduce the search. For example, with M = 200, N = 3000, g = 7, and G_M = 30, G_N = 430 (these numbers are typical of indoor scenes), the search reduction assuming 70% reliability in simple grouping to > 95% reliability in grouping within color regions is $\approx 0.25 * 10^9$ which is a considerable improvement. Consider next, when grouping is restricted to salient color regions. The number of matches further reduces to $O(\sum_{j=1}^{N_S} G_{N_j} G_M g^3 g^3)$, since only the groups in the salient regions need be tried.

To obtain an estimate of the number of matches and time taken for matching in real scenes when color-based selection is used, we recorded the number of regions (obtained by applying the segmentation algorithm of Section 3), and the number of data features within regions in some selected models and scenes (Figures 1 and 2 show typical examples of models and scenes tried). The regions were ordered using the color saliency measure and the four most salient regions were retained. Then search estimates were obtained using the above formulas, and assuming a grouping scheme that gives a number of groups within regions that is bounded

by $\frac{\text{the number of features in a region}}{\text{average size of the groups in a region}}$. This is a good bound on the number of groups produced using simple grouping schemes such as grouping 'g' closely-spaced parallel lines in the region. The result of such studies is shown in Table I. As can be seen from this table, the number of matches is always smaller when salient color regions are used for selection. But the ultimate utility of such a selection mechanism can be accurately gauged only after it is integrated with a recognition system. Current research is being directed towards this effort.

5. COLOR-BASED MODEL-DRIVEN SELECTION

The previous section described a data-driven selection mechanism that was meant for an object of interest having some salient color regions. This will not be of much help when the object of interest is not salient in color (but salient in some other domain, say texture) or is not salient at all. In such cases, the color description of the model can be used to perform selection. We now describe one such color-based model-driven selection mechanism. Here, given a color-based description of a model object, the task is to locate color regions that satisfy this description. The use of model information to constrain the matching of model features to image features is not new. Several model-driven search restriction techniques such as generalized Hough transforms [17], heuristic termination [12], and focal features have evolved [2, 1, 3]. The emphasis in these methods was on geometric constraints that can prune the search space during the matching stage of recognition. The approach we present here, on the other hand, emphasizes some global relational information about model color regions to prune the search space prior to matching. It also provides possible correspondences between model and image regions. Such a correspondence can further reduce the complexity of recognition because the search for pairing model features to data features can be restricted now to these corresponding regions rather than all image regions. Color information in the model object has been used before to search for instances of the object in the given image of a scene [31, 33]. These approaches represent model and image color information by color histograms and perform a match of the histograms. Such approaches usually cause a lot of false positive identifications, and do not explicitly address some of the problems that arise in going from a model object to its instance in a scene. Also, since they do not supply correspondence between model and image regions, they are not as useful for reducing the search in recognition.

In order for any scheme for model-driven selection to be effective for reducing the search in recognition, it must meet two requirements: (i) it must be sufficiently selective to avoid many false positive identifications that cause needless search for matches, and (ii) it must be sufficiently conservative to avoid many false negatives, causing recognition to fail when it should have succeeded. A selection scheme can make false negatives if it does not adequately take into account the various

problems that arise in going from a model object to its image in the scene. An object may not appear the same in the scene as it does in the model, because it has undergone pose changes, or because it is occluded, or its colors appear different in the current illumination conditions. In addition, artifacts such as specularities, inter-reflections, and shadows may also cause changes in the appearance of the object. So how can a model-driven selection mechanism meet these two apparently conflicting requirements? We now describe an approach to model-driven selection that meets some of these requirements. It makes a particular choice of model description and assumes that this is made available to it for selection. Since this model description affects the way our approach formulates the color-based model-driven selection problem, it is described first.

5.1 Model Description

The color region information in the model[3] (in an image or view of the model, that is) is represented as a region adjacency graph (RAG)

$$M_G =< V_m, E_m, C_m, R_m, S_m, B_{rm}, B_{sm} > \tag{12}$$

where V_m = color regions in the model, E_m = adjacencies between color regions, $C_m(u)$ = color of region $u \in V_m$, $R_m(u,v)$ = relative size of region 'v' with respect to region u. $S_m(u)$ = size of region u, and B_{rm} = a bound on the relative size of regions given by R_m, and B_{sm} = a bound on the absolute size of regions given by S_m.

The above description exploits features of regions that tend to remain more or less invariant in most scenes where the model appears. If the color of a model region is specified by its color category, then as we discussed before, it tends to remain relatively stable (or changes in a predictable way) under variations in illumination conditions, and pose changes. Similarly, the adjacency information between two color regions tends to remain more or less invariant in the different appearances of the object, as long as the two regions are visible in the given image and there are no occlusions. Finally, the relative size of regions is preserved under changes of scale. But it can undergo considerable changes if the pose of the object changes, say when a region goes partially out of view. The bound on the relative size changes in each pair of adjacent region, B_{rm} indicates the extent of pose changes that a selection mechanism is expected to tolerate. Relative size changes can also occur due to occlusions. By placing some loose bounds on the absolute size changes as given by B_{sm}, the model description restricts the changes that can be tolerated in the presence of occlusions. For size changes in a region that go beyond the bounds,

[3]The model description specifies a color view, that is, a range of 2D views of the model in which one or more of the color regions described in the model are visible. If the model has some views showing an entirely different set of color regions, then they must be specified as separate color views.

that region will be considered no longer recognizable, and then the selection will have to depend on the evidence for other model regions in the image.

This description is fairly rich and has some structural information about color regions that can be used to restrict the number of false positives, and some constraints on the relative and absolute size changes that can be used to restrict the number of false negatives made by the selection mechanism.

Finally, the model description gives a way to analogously organize the color region information in the image as an image region adjacency graph as $I_G = < V_I, E_I, C_I, R_I, S_I >$, where each term has a meaning analogous to $< V_m, E_m, C_m, R_m, S_m >$ respectively.

5.2 Formulation of the Color-based Model-driven Selection Problem

In this section we will formulate the color-based model-driven selection problem as a type of subgraph matching problem. Given the image region adjacency graph, the model object if present in the scene represented in the image will form a subgraph in I_G. The location strategy can be regarded as the problem of searching for suitable subgraphs that satisfy the model description. Any such subgraph $I_g =< V_g, E_g, C_g, R_g, S_g >$ such that $\|V_g\| \leq \|V_m\|, \|E_g\| \leq \|E_m\|$, has associated with it a node correspondence vector $\Upsilon = \{(u_m, u_g) | \forall u_m \in V_m, u_g \in V_g \cup \{\perp\}, \{\perp\}$ is a null match$\}$. Although there are an exponential number of such subgraphs, not all of them correspond to model RAG. From the model description a set of unary and binary constraints could be derived (as is described later) that make only some subgraphs feasible. A feasible subgraph is, therefore, a subgraph that has all its nodes satisfying unary and binary constraints. For model-driven selection, since it is desirable to have at most one image subgraph matching the model RAG, we can select from among these subgraphs, a subgraph(s) that in some sense best satisfies the model description. Here we formulate color-based model-driven selection as the problem of choosing a feasible subgraph(s), I_g that minimizes the following measure:

$$\text{SCORE}(I_g) = (1 - \frac{\|V_g\|}{\|V_m\|}) + \frac{2 \sum_{\forall (u_g, v_g) \in E_g, \Upsilon(u_m)=u_g, \Upsilon(v_m)=v_g} R^2_{mg}(u_m, v_m, u_g, v_g)}{\|E_m\|}. \tag{13}$$

where $R_{mg}(u_m, v_m, u_g, v_g)$ expresses the change in the relative size when adjacent model regions (u_m, v_m) are paired to corresponding image regions (u_g, v_g) and is given by $R_{mg}(u_m, v_m, u_g, v_g) = \frac{|R_m(u_m, v_m) - R_g(u_g, v_g)|}{\max(R_m(u_m, v_m), R_g(u_g, v_g))}$. SCORE$(I_g)$ emphasizes rewards for making as many correspondences as possible as indicated by the first term, called Match(I_g), and penalties for a mismatch of the relative size, as indicated by the second term, called Deviation(I_g), which measures the mean square deviation of the relative sizes. Since the subgraphs are all feasible, the deviation accounts for occlusions and pose changes in a more refined way than the binary

constraints alone. Another advantage of this measure is that it can be incrementally computed from individual region matches, so that a branch-and-bound search formulation can be used to reduce considerably the search involved in finding the best subgraph (i.e. the one with the lowest score). Finally, the above formulation is based on the hypothesis that at least one of the regions in the isolated subgraph corresponds to a model region. It is also designed primarily to locate single instances of the model object in the image. More instances can be found after removing the regions in the found instance from the image RAG.

5.3 A Color-based Model-driven Selection Mechanism

A color-based model-driven selection mechanism was built using the above formulation. The mechanism essentially uses a search strategy to find the best subgraph. The result of selection is the correspondence vector associated with the best subgraph. The search strategy used the following constraints to restrict the search among feasible subgraphs.

1. Unary constraints: The color and absolute region size information provided in the model description were used to develop unary constraints on these features. The color $C_g(u_g)$ of an image region u_g is said to match the color $C_m(u_m)$ on a model region u_m if these colors belong to the same category or compatible categories (described in Section 2.4). With this scheme, brighter colors (of a given hue) in the model could potentially match to darker colors of the same overall hue in the image, thus accounting for simple lowering in illumination levels. The bounds on the absolute size provided by B_{sm} act as loose size constraints to rule out some clearly absurd scale changes (such as, say, a 100 fold increase in the smallest model region implying a blowup of the model outside the image bounds).

2. Binary constraints: The adjacency (as well as non-adjacency) and relative size information provided in the model were used as binary constraints to prune some impossible subgraphs. Specifically, the lack of adjacency in model regions is a powerful constraint, because two adjacent regions in the image cannot correspond to two regions that are not adjacent in the given color description (assuming a rigid model)[4]. Two adjacent regions in the model may, however, not appear adjacent in a given image due to occlusion. A simple analysis of occlusions could rule out several false matches in such cases (such as, say, discarding a match if the area spanned by the occlusion within a rectangle enclosing the candidate non-adjacent image regions far exceeds the combined size of the corresponding adjacent model regions). The bound on the relative sizes served as another binary constraint. The bound B_{rm} was used to constrain possible matches by requiring $R_{rg}(u_m, v_m, u_g, v_g) \leq B_{rm}(u_m, v_m)$.

3. Searching for the best subgraph

The search for the best subgraph (i.e. the subgraph that minimize the value

[4]Notice here that the search is for a given color view of the model.

of SCORE), can in principle, be done by an exhaustive enumeration of subgraphs. But with the algorithm described below, the search required is reduced to a large extent. The algorithm used is essentially a variation of the branch and bound interpretation tree (IT) search [12], with the major difference being that no verification is done when the search reaches a leaf node (as the task is selection and not recognition). Each level of the search tree represents a possible match for a model region (this includes a null match), so that the depth of the search tree is fixed by the number of nodes in the model RAG. The unary constraints are checked *a priori* to prune the breadth of the search tree. A subgraph in the image RAG that is a potential match for the model RAG is represented by a path in the IT. The value of SCORE is updated at each node as $\text{SCORE}_{i+1} = \text{SCORE}_i - \frac{1}{\|V_m\|} + \frac{2R^2_{mg}}{\|E_m\|}$. By keeping the lowest value of SCORE so far, search can be cut off below any node with a Deviation(I_g) value greater than the lowest SCORE value. In practice, the unary and binary constraints prune the search tree considerably so that the average number of full paths (up to the leaves) explored are few (≈ 50). Finally, after an instance of the model region has been found in the image, the selected area is removed and the search repeated on the resulting image RAG to look for more instances of the model object.

5.4 Results

The result of using color-based model-driven selection are illustrated in Figures 2 and 3. Figure 2a shows a model object, and its color description obtained by using the color-region segmentation algorithm of Section 3 is shown in Figure 2b. Here the background was removed by a simple threshold on intensities. This description is used to create a model RAG. Figure 2c shows a scene in which the model object occurs. The scene shown has several other objects with one or more of the model colors. Also, the model appears in a different pose here, being rotated to the left about the vertical axis. Figure 3b shows the result of applying the unary color constraints. The big blue glass matches the small blue flowers based on color alone. Next, the unary constraint on absurd size changes are used to prune the possibilities and the result is shown in Figure 3c. Finally, the subgraph with the lowest value of SCORE is shown in Figure 3d. As can be seen from this figure, a region containing most of the model object has been identified even though the color image segmentation was not perfect (notice the small streak above the white rim of the cup that merges with the book in the background).

5.5 Search Reduction using Color-based Model-driven Selection

The color-based model-driven selection mechanism provides a correspondence of model regions to some image regions. The matching of model features to image features can be restricted to within corresponding regions, and this reduces the number of matches that need to be tried for recognition. To reduce the search

further, conventional grouping can be performed within the selected color regions, as described in Section 4.2. To estimate the search reduction in this case, we continue with the analysis done in that section. Let N_l be the number of solution subgraphs given by the selection mechanism, and let I_k represent one such subgraph with the number of nodes = N_k. Let (G_{u_j}, G_{v_i}) = the number of groups in region u_j of the solution subgraph I_k, and region v_i of the model RAG that corresponds to u_j as implied by the correspondence vector Υ associated with I_k. Then assuming, as before, the average size of the group = g, the number of matches that need to be tried are $O(\sum_{k=1}^{N_l} \sum_{j=1}^{N_k} G_{u_j} G_{v_i} . g^3 . g^3)$. To compare this kind of selection with pure grouping we can take some typical values of these numbers. Letting M = 200, N = 3000, g = 7, G_M = 30, G_N = 430, G_{u_j} = 8, G_{v_i} = 5, N_l = 5, N_k = 5, we have the number of matches with grouping alone to be $O(G_M G_N g^3 g^3) \approx 1.56 * 10^9$, and using model-driven color-based selection with grouping, the number of matches become $\approx 1.25 * 10^8$. Assuming 1 microsecond as time per match this corresponds to reduction in match time from 26 minutes to $\approx$ 2 minutes. By trying several models and images of scenes where they occured, we recorded the average number of subgraphs generated by the model-driven selection mechanism. The search estimates were obtained using the above formula for model-driven selection with grouping, and the formulas for other methods mentioned in Section 4.2. The results are shown in Table II. The bound on the number of groups in a region was the same as used in Section 4.2. As can be seen from the table, the number of matches using correspondence between model and image color regions is always lower. A curious feature to note from the table is that it takes less number of matches (and hence lesser time) for a more complex model (entry 1 in Table II) containing several color regions, than for a simple object with fewer regions (entry 2 in Table II). This is understandable since, with a large number of regions, the constraints are stronger and hence the false matches are fewer.

Discussion: The above studies estimated the search reduction without actually integrating the selection mechanism with a recognition system. Moreover, the estimated search was based on the assumption that there were no false negatives given by the selection mechanism. This can happen since a subgraph with the lowest value of SCORE may not always indicate a match to the model. To estimate the number of false positives, the number of false negatives, and the reduction in search that results due to this color-based selection mechanism, we have recently developed a 3D from 2D recognition system and are currently testing it. Preliminary results on using the selection mechanism as a front-end for recognition have so far been encouraging.

6. SUMMARY

In this paper we have shown how color can be used as a cue to perform both data and model-driven selection. Unlike other approaches to color, we have used

the intended task to constrain the kind of color information to be extracted from images. This led to a fast color image segmentation algorithm based on perceptual categorization of colors to given perceptually different color regions. This color description of the image formed the basis of data and model-driven selection. A saliency measure was then developed to rank the color regions to perform data-driven selection. Lastly, an approach to model-driven selection was presented that exploited description of model color regions to locate instances of model in the image. Finally, we regard color as one of the many cues that can be used for selection. Future research is directed towards using other cues such as texture to perform data and model-driven selection.

APPENDIX A

In this appendix we describe the psychophysical experiments done to derive the color categories. The aim of these experiments was to record the perceptual judgements of colors in different regions of the color space by a systematic exploration of the color space. For this, the hue-saturation-value representation of color space was used. As shown in Figure 7, the entire spectrum of computer recordable colors (2^{24} colors) was quantized into 7200 bins corresponding to a 5 degree resolution in hue, and 10 levels of quantization of saturation and intensity values. In order to scan the color space systematically, the colors in bins were observed starting with the bins of red hue and going around the color space back to the red hue again. The display set up involved a 24-bit high resolution monitor with appropriate monitor calibration to observe the colors in dark room conditions with a minimum viewing distance of 2 feet. Uniform color samples (mondrians) of size 64 x 64, corresponding to the hue-saturation-and brightness value in each bin were displayed on the screen. The set of mondrians displayed on the screen varied in purity vertically, and intensity horizontally, while the hue was kept constant. For each hue the colors initially displayed had a resolution of 0.2 in brightness and saturation. Four subjects were tested individually and were supplied with a chart that showed the gradations in brightness and purity varying in a manner that corresponded to the color spectrum shown on the display. Each subject was then asked to group the color samples displayed on the screen into perceptually uniform color groups and mark the result on the chart provided, so that the end result was a segmentation of the chart into perceptually uniform colored groups. The presence of a boundary was taken to mark a change in color category. To precisely locate this boundary, the color samples around the boundary were redisplayed with a finer resolution (of 0.1) in brightness and saturation. Before assigning a new category label each group is compared with groups of previous hue by displaying the colors in the previous group along with a given group and asking the subject to judge if this group could be merged with the previous hue groups. The observation of successive mondrians was done with a 10 minute intervals in between to remove after-effects of the

previous display. The mondrians displayed were sufficiently apart on the screen to keep the effects of simultaneous contrast small. By averaging out the differences in the responses between subjects, we found about 220 different color categories were sufficient to describe the color space. The color category information was then summarized in a color-look-up table.

References

[1] N. Ayache and O.D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," IEEE Trans. Pattern Anal. and Machine Intell., vol. 8, no.1, 1986, pp. 44-54.

[2] R.C. Bolles and R.A. Cain, "Recognizing and locating partially visible objects: The local feature focus method," Int. Jl. Robotics Res., vol.1, no.3, 1982, pp. 57-82.

[3] R.C. Bolles and P. Horaud, "3DPO: A three dimensional part orientation system," Int. Jl. Robotics Research, vol. 5, no.3, 1986, pp. 3-26.

[4] M.H. Bornstein, W. Kessen, and S. Weiskopf, "Color vision and hue categorization in young human infants", Jl. of Exper. Psychol: Human Perception and Performance, vol.2, no.1, 1976, pp.115-129.

[5] E. Carterette and M. Friedman, *Perceptual Coding*, New York: Academic Press, 1978.

[6] D. Clemens, "Region-based feature interpretation for recognizing 3D models in 2D images," TR-1307, Ph.D Thesis, M.I.T. Artificial Intell. Lab., 1991.

[7] D.T. Clemens and D.W. Jacobs, "Space and time bounds on indexing 3D models from 2D images," IEEE Trans. Pattern Anal. and Machine Intelligence, vol.13, Oct. 1991, pp. 1007-1017.

[8] M.G. Engel, "Visual conspicuity and selective background interference in eccentric vision," Vision Research, vol.14, 1974, pp.459-471.

[9] J.D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Reading: Addison Wesley, 1984.

[10] B. Funt and J. Ho, "Color from black and white", in Proc. Second Int. Conf. on Computer Vision, Tampa, Florida, 1988, pp.2-8.

[11] R. Gershon, A. D. Jepson, J.K.Tsotsos, "From R,G,B to surface reflectance: Computing color constancy descriptors in images," in Proc. 10th Int. Jt. Conf. on Artificial Intelligence, 1987, pp. 755-758.

[12] W.E.L.Grimson., *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press: Cambridge, 1990.

[13] B.K.P.Horn, *Robot Vision*, MIT Press: Cambridge, 1986, Chapter 4.

[14] A. Hurlbert, *The Computation of Color*, Tech. Rep. TR-1154, Artificial Intell. Lab., MIT, 1989.

[15] A. Hurlbet and T. Poggio, "Visual attention in brains and computers", AI Memo. 915, Artificial Intell. Lab., M.I.T., June 1986.

[16] D. Huttenlocher and S. Ullman, "Object recognition using alignment," First Int. Conf. Computer Vision, London, 1987, pp.102-111.

[17] J. Illingworth and J. Kittler, "A survey of the Hough transform", Comp. Vision, Graphics, Image Proc., vol.44, 1988, pp. 87-116.

[18] D.W. Jacobs, "Grouping for recognition," AI Memo. 1177, M.I.T. Artificial Intell. Lab., 1989.

[19] D.G. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academic: Boston, 1985.

[20] E. Kandel and J. Schwartz, *Principles of Neural Science*, Elsevier: New York, 1985, Chapter 30, pp.386-388.

[21] G.J. Klinker, S.A. Shafer, T. Kanade, "Using a color reflection model to separate highlights from object color," Proc. Int. Conf. Computer Vision, June 1987.

[22] G.J. Klinker, S.A. Shafer, and T. Kanade, "A physical approach to color image understanding," Intl. Jl. Computer Vision, vol.4, no.1., pp.7-38, Jan. 1990.

[23] E. Land, " Recent advances in retinex theory," in *Central and Peripheral Mechanisms of Colour Vision*, T. Ottoson and S. Zeki Ed., pp. 5-17, London:McMillan, 1985.

[24] J. Mahoney, "Image chunking: Defining spatial building blocks for scene analysis," Technical Rep. TR-980, Artificial Intell. Lab., MIT, 1986.

[25] L.T. Maloney and B. Wandel, "Color constancy: A method for recovering surface spectral reflectance," Jl. Optical Society of America, vol.3, 1986, pp.29-33.

[26] G.W. Meyer and D.P. Greenberg, "Perceptual color spaces for computer graphics," Computer Graphics, vol.14, no.3, July 1980, pp. 254-261.

[27] E. Rosch, "The nature of mental codes for color categories," Jl. of Exper. Psychol: Human Perception and Performance, vol.1, no.4, pp. 303-322, 1975.

[28] D.Sagi and B. Julesz, ""where" and "what" in vision", Science, vol.228, 1985, pp.1217-1219.

[29] A. Shashua and S. Ullman, "Structural saliency : The detection of globally salient structures using a locally connected network," in Proc. 2nd Int. Conf. Computer Vision, pp. 321-327, Florida, Dec. 1988.

[30] E. Sternheim and R. Boynton, "Uniqueness of perceived hues investigated with a continuous judgemental technique," Jl. of Experimental Psychology, vol.72, pp.770-776, 1966.

[31] M.J. Swain and D. Ballard, "Indexing via color histograms," in Proc. Third Int. Conf. Computer Vision, 1990, pp. 390-393.

[32] M. Tsukada and Y. Ohta, "An approach to color constancy using multiple images," in Proc. Third Int. Conf. on Computer Vision, 1990, pp. 385-389.

[33] L.E. Wickson and D. Ballard, "Real-time detection of multi-colored objects," in SPIE Sensor II: Human and Machine Strategies, vol.1198, Nov. 1989.

[34] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley & Sons, New York, NY, 1982.

Figure 1: Illustration of color region segmentation and color-saliency. (a) Input image depicting a scene of objects of different materials and having occlusions and inter-reflections. (b) Segmented image using the color region segmentation algorithm. (c)–(f) The four most distinctive regions detected using the color-saliency measure. The white portion in the red book appears so because of the white background.

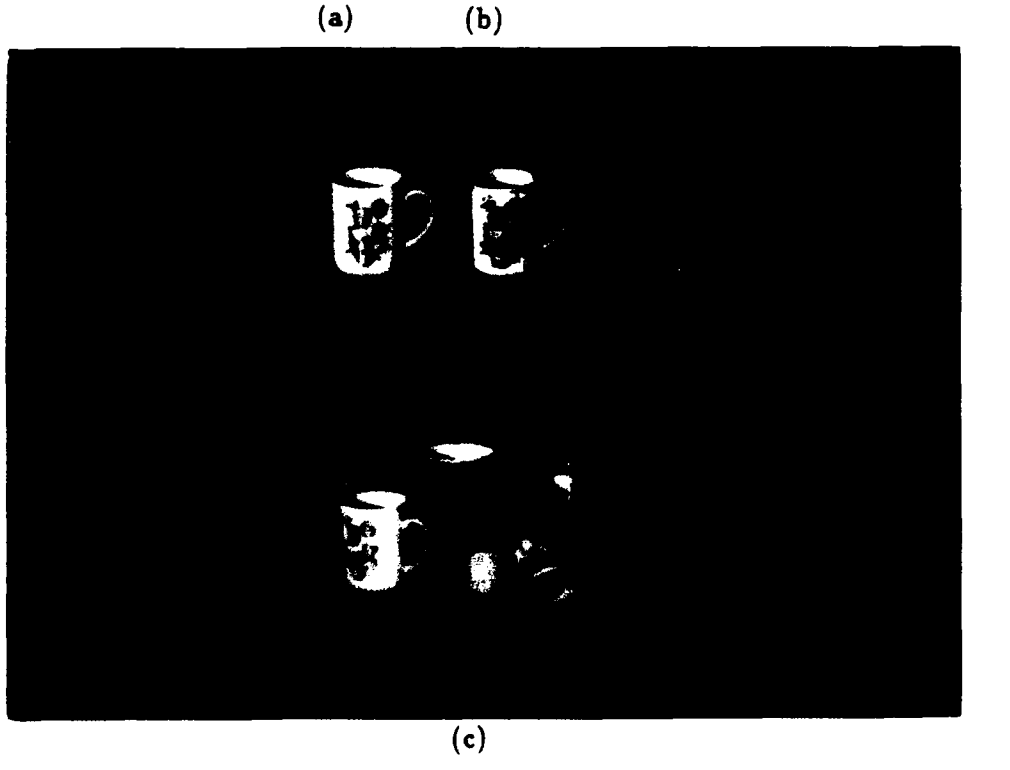


Figure 2: Illustration of model-driven selection — Model and scene. (a) The object serving as the model. (b) Its color description produced by the segmentation algorithm of Section 3. (c) A cluttered scene in which the object appears.

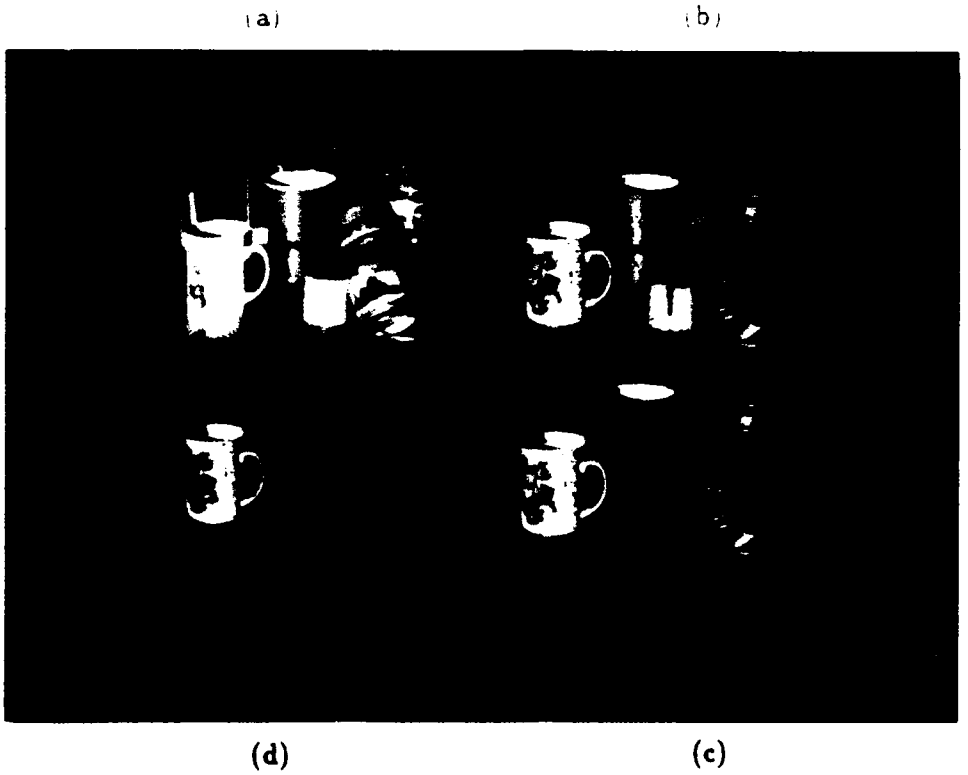


Figure 3: Illustration of color-based model-driven selection. (a) A scene containing the model object of Figure 2a. (b) Regions selected based on unary color constraint. (c) Regions of (b) pruned after using the unary size constraint. (d) Regions corresponding to the best subgraph that matched the model specifications.

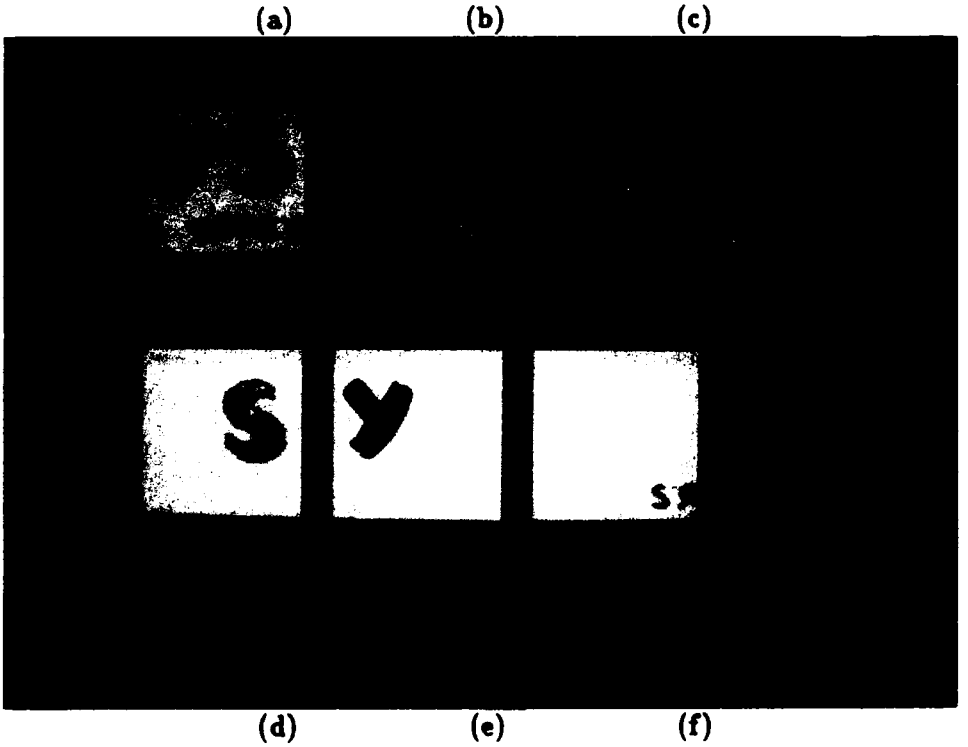
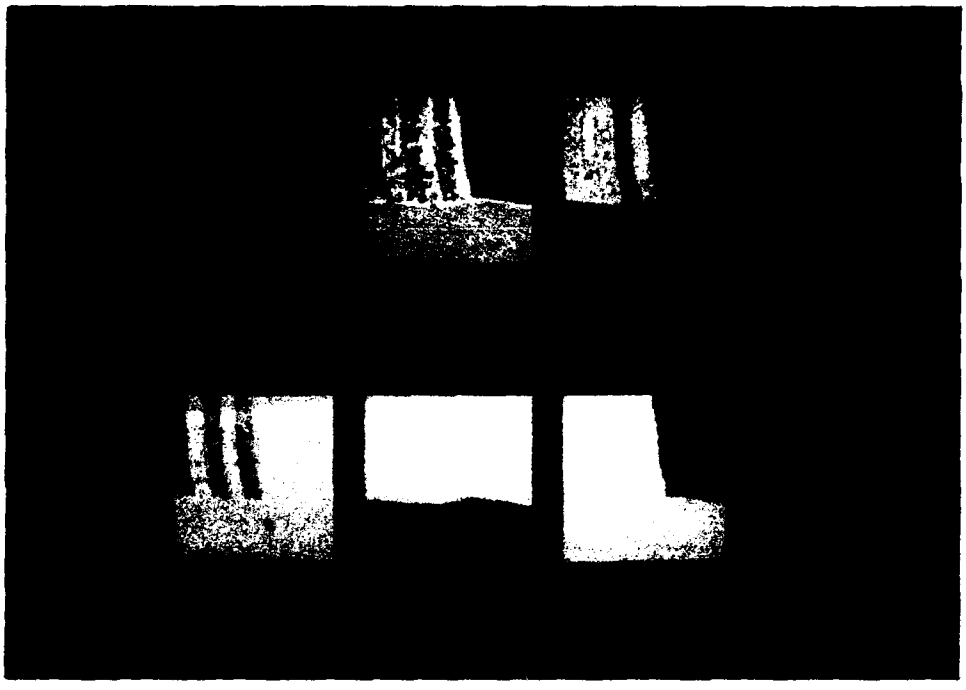


Figure 4: Illustration of color region segmentation and color-saliency. (a) Input image consisting of regions of 3 different colors: red, green and blue against an almost white background. (b) Result of step2 of algorithm with regions colored differently from the original image. (c) Final segmentation of the image of Fig.3a. (d) — (f) The three most distinctive regions found using the color saliency measure.

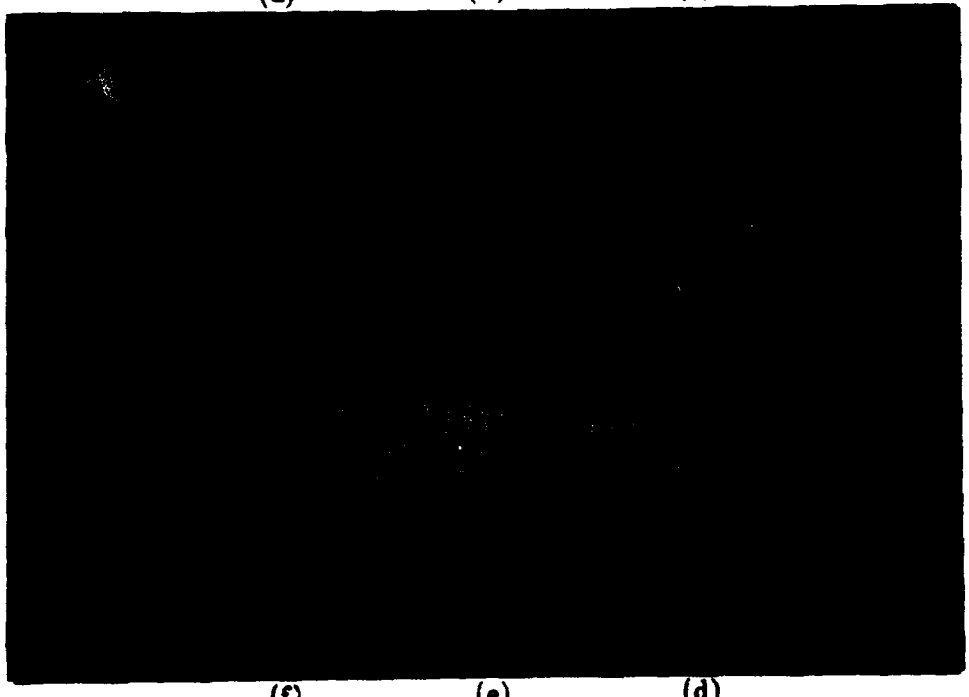
(a) (b) (c)



(d) (e) (f)

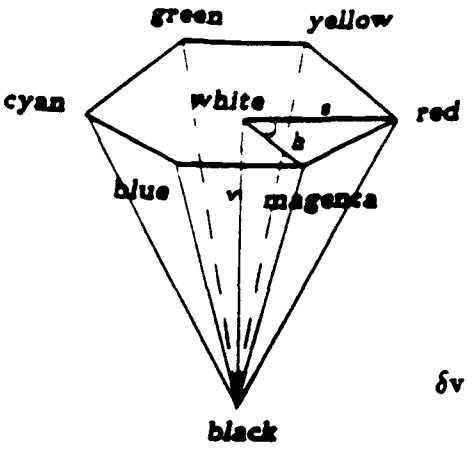
Figure 5: Illustration of color region segmentation and color-saliency — Another example. (a) Input image of a set of colored cloth materials. (b) Regions obtained at the end of step-2 of algorithm (before merging overlapping regions). (c) Final segmented image suitably recolored to show the segmented regions. (d) - (f) The three most distinctive regions found using the color saliency measure.

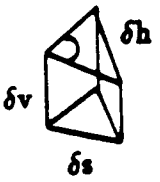
(a) (b) (c)



(f) (e) (d)

Figure 6: Illustration of color region segmentation and color-saliency — Last example. (a) Input image depicting a scene of different kinds of objects (cloths and polished book). (b) The color regions extracted from (a) using the color region segmentation algorithm. (c)-(f) The four most distinctive regions detected using the color-saliency measure.

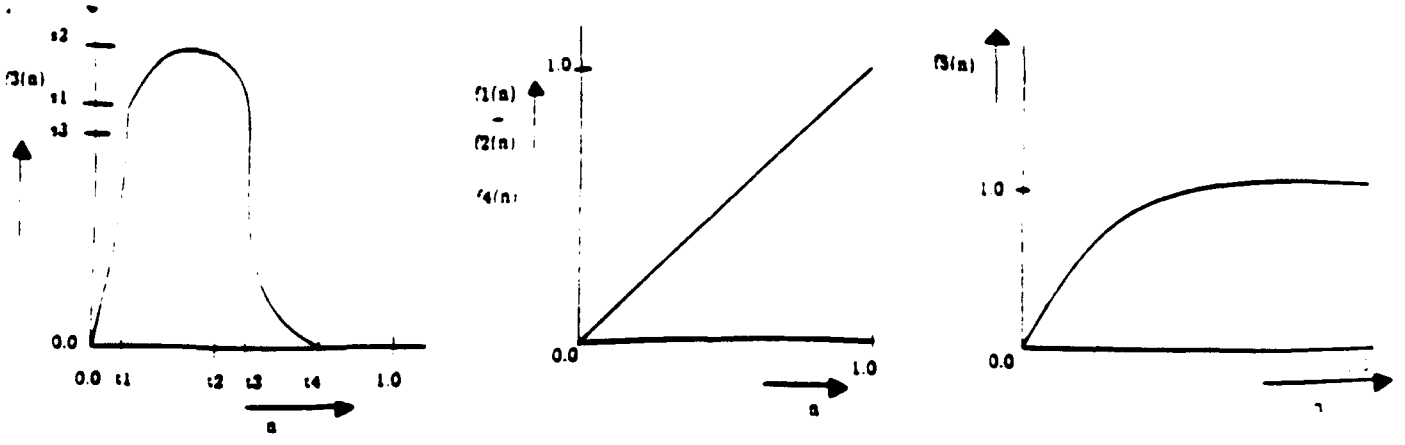




δh	δs	δv	number of bins	number of categories
5	0.1	0.1	7200	220

(a) (b) (c)

Figure 7: Illustration of the quantization of the hsv-color-space. (a) hsv-color model. (b) a cell of the quantized color space. (c) The quantization data and the number of categories obtained.



(a) (b) (c)

Figure 8: Graphs of weighting functions used in devising the color-saliency measure.

S.No	M	N	M_R	N_R	No selection		Only grouping		Salient color + grouping	
					Num. matches	Time	Num. matches	Time	Num. matches	Time
1.	229	1170	1	18	1.92×10^{16}	610yrs	6.52×10^{8}	11min	3.37×10^{8}	5min
2.	507	2655	2	20	2.4×10^{18}	77,341yrs	3.22×10^{9}	54min	1.32×10^{9}	22min
3.	124	2655	2	20	3.57×10^{16}	1131yrs	8.05×10^{8}	13min	3.3×10^{8}	5min
4.	507	2247	2	14	1.48×10^{18}	46,884yrs	2.72×10^{9}	46min	7.8×10^{8}	13min

Table I: Search reduction using color-based data-driven selection. The last column shows the match time when color-based data-driven selection is combined with grouping. The color-based selection is done by choosing the four most salient regions. Here g = 7, Time per match = 1 microsecond, and the grouping method is as described in text.

S.No	M	N	M_R	N_R	N_l	N_k	No selection		Only grouping		Model-driven selection	
							Num. matches	Time	Num. matches	Time	Num. matches	Time
1.	786	3268	5	30	1	(3)	1.69×10^{9}	530000yrs	6.15×10^{8}	103min	4.55×10^{7}	45sec
2.	83	3078	1	20	3	(1,1,1)	1.67×10^{16}	528yrs	6.2×10^{8}	11min	1.7×10^{8}	3min
3.	507	2655	2	20	2	(2,1)	2.4×10^{18}	77,341yrs	3.22×10^{9}	54min	3.72×10^{8}	6min
4.	507	2247	2	14	1	(2)	1.48×10^{18}	46,884yrs	2.72×10^{9}	46min	3.16×10^{8}	5min

Table II: Search reduction using color-based model-driven selection. The last column shows the match time when model-color-based selection is combined with grouping. Here g = 7, Time per match = 1 microsecond, and the grouping method is as described in text.